

Tiva Based Daughter Board for Firebird V Hardware And Software Manual.

eRTS Lab IIT Bombay

June 17, 2017

1 Credits

Version 1.0

June 17, 2017

Documentation Author(Alphabetical Order):

1. Ayush Gaurav, Intern eYSIP 2017
2. Nagesh K, Intern eYSIP 2017

Credits(Alphabetical Order):

1. Prof Kavi Arya, CSE IIT Bombay
2. Nex Robotics Pvt. Ltd.
3. Piyush Manavar, Team e-Yantra

2 Notice

The contents of this manual are subject to change without notice. All efforts have been made to ensure the accuracy of contents in this manual. However, should any errors be detected, e-yantra welcomes your corrections. You can send us your queries / suggestions at [Contact Us](#)

Table Of Content

1	Credits	2
2	Notice	3
3	Introduction	6
4	Tiva Based Daughter Board	6
4.1	Technical Specification	6
5	Hardware Manual:	7
5.1	Voltage Regulation on the Daughter Board	7
5.1.1	Powering Micro-controller	7
5.1.2	Powering Servos	7
5.2	Level Converters	8
5.3	Sensors	8
5.3.1	3.3V sensors	9
5.3.2	5V sensors	9
5.4	Port Expander	10
5.5	External ADC	10
5.6	LCD Interfacing	11
5.7	USB Communication	12
5.8	Programing the Controller	13
5.9	Reset Switch	13
5.10	Servo Connectors	13
5.11	TM4C123GH6PM Micro-controller:	14
5.12	Pin Functionality	17
6	Software Manual:	18
6.1	Code Composer Studio:	18
6.1.1	Download CC Studio:	18
6.1.2	Installing C C Studio:	18
6.1.3	Create a New Project	20
6.1.4	Add Path and Build Variables	20
6.2	driver.lib	25
6.3	Buzzer	25
6.4	Programming the Robot	26
6.5	Using Debugger of The Programmer	26
6.6	Simple I/O Operation	26
6.7	Robot Direction Control	26
6.8	Robot Position Control Using Interrupts	26

6.9	Timers and its Interrupts	26
6.10	Robot Speed Control	26
6.11	LCD Interfacing	26
6.12	Analog To Digital Converter	26
6.13	Serial Communication	26
6.14	I2C Communication	26

3 Introduction

4 Tiva Based Daughter Board

There are two daughter boards one with the launchpad and other one with the Arm Cortex M4 based uC. Almost all the specification are same unless mentioned otherwise.

4.1 Technical Specification

Microcontroller:

TM4C123gh6pm (ARM architecture based Microcontroller)

To know more about the microcontroller please refer to [datasheet](#).

Sensors:

Three white line sensors (extendable to 7)

Five Sharp GP2Y0A02YK IR range sensor (One in default configuration)

Eight analog IR proximity sensors

Two position encoders

Indicators:

2 x 16 Characters LCD

Buzzer

Communication:

USB Communication

Wireless ZigBee Communication (2.4GHZ) (if XBee wireless module is installed)

Bluetooth communication (Can be interfaced on external UART0 available on the board)

Simplex infrared communication (From infrared remote to robot)

I2C Communication

Battery Life:

2 Hours, while motors are operational at 75% of time

Locomotion:

Two DC geared motors in differential drive configuration and caster wheel at front as support

Top Speed: 24 cm / second

Wheel Diameter: 51mm

Position encoder: 30 pulses per revolution

Position encoder resolution: 5.44 mm

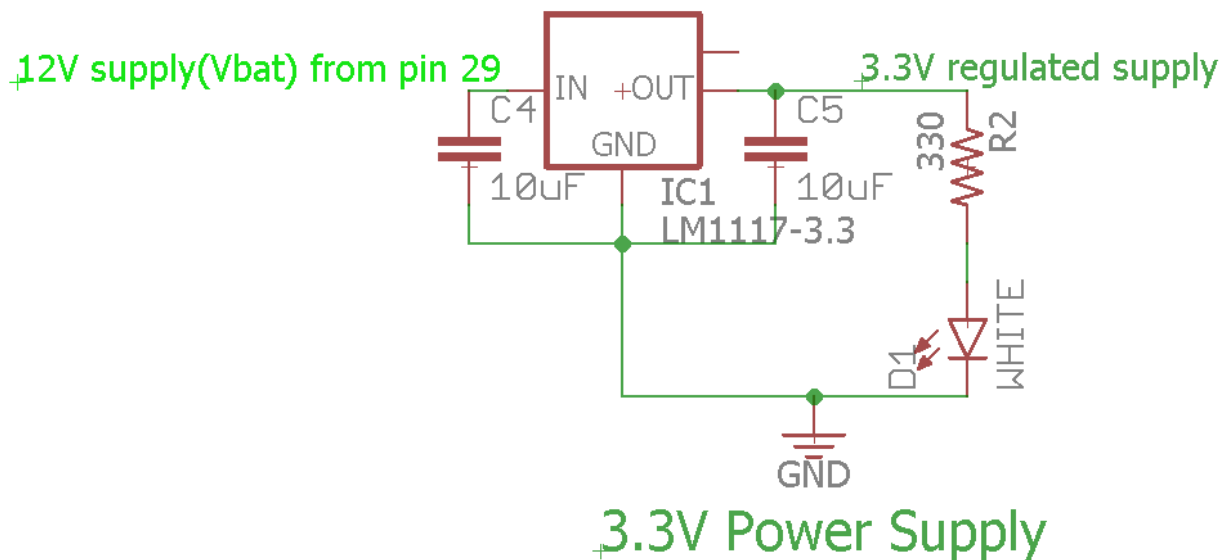
5 Hardware Manual:

5.1 Voltage Regulation on the Daughter Board

The voltage source available on the Firebird is 9.6V. But the TIVA platform works on 3.3V and the servos can operate upto 6V. So there must be 3 different voltage levels on the board. The uC based board has 2 voltage regulators and the plug and play board has 1 voltage regulator. In the uC based board the 9.6 volts is 3.3V to power the microcontroller. In the plug and play board there is an inbuilt voltage regulator, so it is directly connected to 5v, 300mA source. The servo in both the boards has a separate 5V regulator.

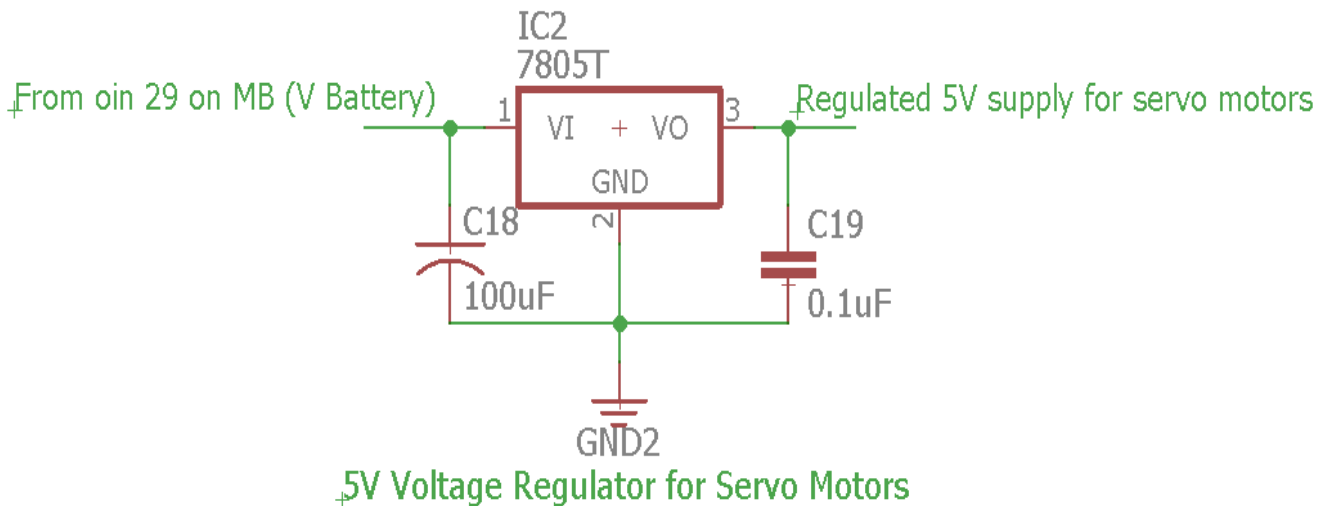
5.1.1 Powering Micro-controller

The boards have different powering circuits. In the plug and play board is connected to 5V source on Pin 10. In the uC based board the 9.6V source available on Pin 29 is reduced to 3.3V. Refer to the schematic below for further details.



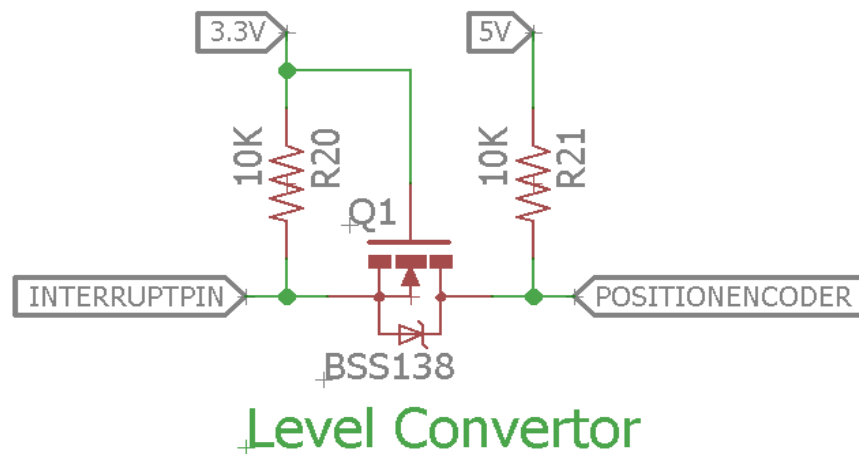
5.1.2 Powering Servos

The servo motors can operate safely up to 6V, beyond this voltage they get damaged. Also, the servos require high current. There is a separate power line for servos taken from Pin 29 and reduced to 5V using the voltage regulator. Refer the schematic for further details.



5.2 Level Converters

The TIVA platform operates at 3.3V and the Firebird operates at 5V. Directly connecting these pins to the TIVA may be fatal. So to interface these sensors, a level converter is used. A bidirectional MOS-FET based level converter is used. The level converter is necessary for input pins. In the boards Level converter is used for interfacing the position encoders of the motors. Refer the schematic for further details.



NOTE: If the user wishes to interface extra sensors using the GPIOs provided on the board, then external level converters have to be used if the output of the sensor is above 3.3V.

5.3 Sensors

The Firebird V has as many as 22 sensors, but maximum 12 sensors can be interfaced directly with the controller. The daughter board has interfaced 20 of those 22 sensors using external I2C based ADC. Sensors that were not included in the daughter board are current sensor and battery monitoring sensor. These sensors are working either on 3.3V or on 5V. Interfacing 3.3V sensors are simple and can be directly connected to the controller. On the other hand 5V can not be directly interfaced so a different approach is taken which will be mentioned in the 5V sensors sub heading.

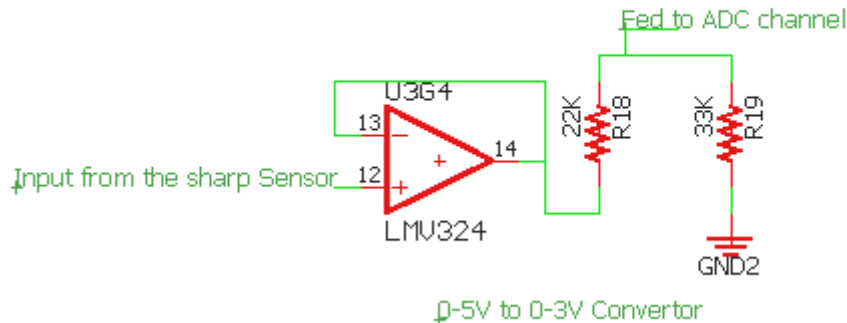
5.3.1 3.3V sensors

The output white line sensors and IR Proximity sensors vary from 0 to 3.3V. Hence these sensors can be interfaced directly with the microcontroller. Refer the table below for pin connections.

IR Proximity Sensors	Pin Name(uC)	Pin Name(Plug and Play)
1	PE1	PB5
2	PE3	PD0
3	PE5	PD3
4	PE4	PD1
5	PB5	PE5
6	External ADC IN6	External ADC IN7
7	External ADC IN7	PE0
8	External ADC IN0	External ADC IN0
White Line Sensors	Pin Name(uC)	Pin Name(Plug and Play)
1	PD2	PE1
2	PD1	PE2
3	PD0	PE3
4	External ADC IN1	External ADC IN2
5	External ADC IN2	External ADC IN3
6	External ADC IN3	External ADC IN4
7	External ADC IN4	External ADC IN5

5.3.2 5V sensors

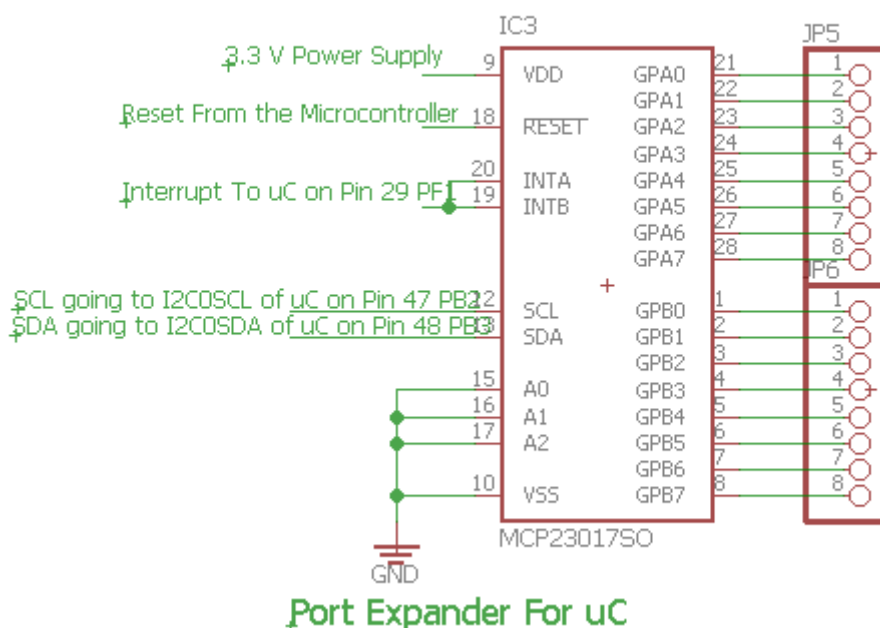
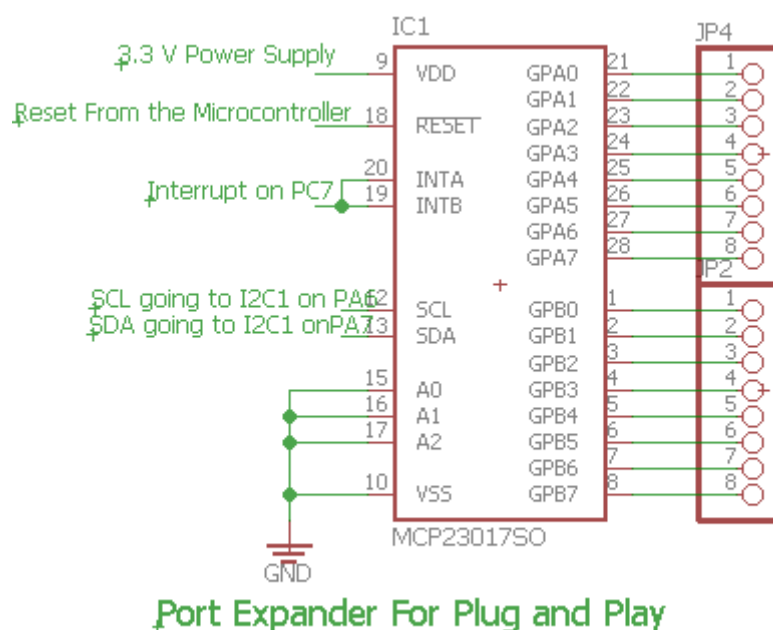
Sharp Sensors are the only sensors on board that works on 5V supply. The output of the sharp sensor ranges from 0-5V and according to the output we have a formula to calculate the distance. While uC has VREF as 3.3V so these sensors cannot be directly connected. The approach we followed is to feed the output of the sensor to a buffer and then using a voltage divider convert 0-5 range to 0-3V range. For better understanding refer to the schematic below. There is a also table which tells about the pin connection.



Sharp Sensors	Pin Name(uC)	Pin Name(Plug and Play)
1	PE0	PB4
2	PE2	External ADC IN1
3	PD3	PD2
4	External ADC IN5	External ADC IN6
5	PB4	PE4

5.4 Port Expander

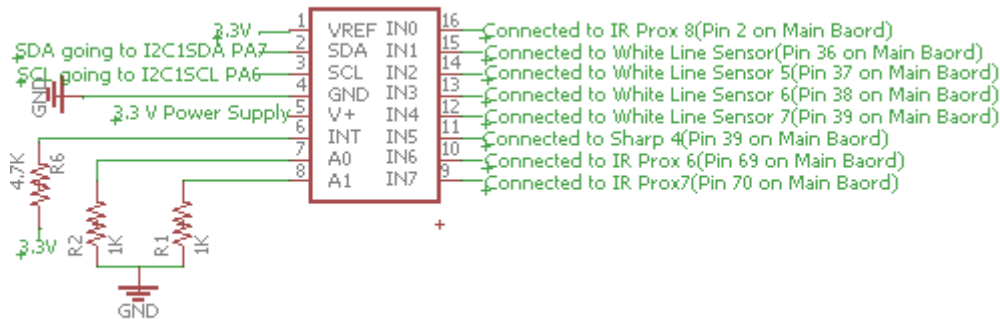
TM4C123GH6PM has only 64 pins out of which only 43 are GPIO pins. This limits our application to read input and respond correspondingly. To increase the number of GPIO and there interrupts we have used I2C compatible a port expander MCP23017. It has 2 PORTS A and B, with each port having 8 Pins. The interrupts on each pin can also be monitored. To read more about it, download the datasheet from [here](#). The schematic of the connection is shown below. Keep in mind that I2C SCL and SDA have already been pulled up using 10K resistor.



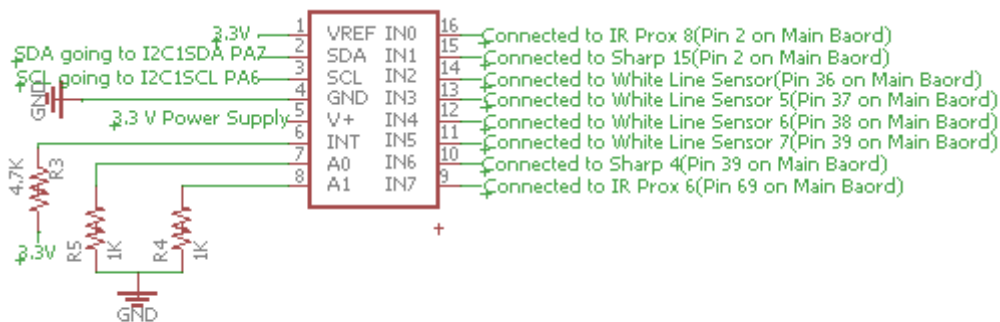
5.5 External ADC

It has already been mentioned that adc channels on the microcontroller is limited to 12 while firebird has

22 sensors available. We have interfaced an external ADC which is also I2C compatible. It has 8 channel with 12 bit resolution. To read more about it, download the datasheet from [here](#). The schematic of the connection is shown below. Keep in mind that I2C SCL and SDA have already been pulled up using 10K resistor.



External ADC connection for uC Board



External ADC connections for Plug and Play Board

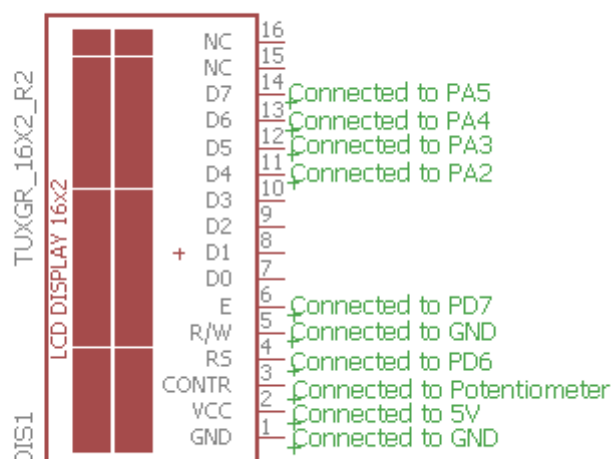
5.6 LCD Interfacing

LCD can be interfaced in 8bit or 4 bit interfacing mode. In 8 bit mode it requires 3 control line and 8 data lines. To reduce number of I/Os required, Fire Bird V robot uses 4 bit interfacing mode which requires 2 control lines and 4 data lines. In this mode upper and lower nibble of the data/command byte needs to be sent separately. RW(Read/Write) control line of lcd is grounded so it can only work in write mode.

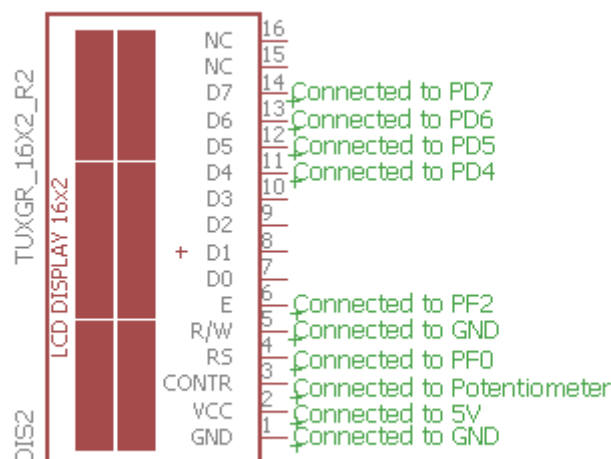
The EN line is used to tell the LCD that microcontroller has sent data to it or microcontroller is ready to receive data from LCD. This is indicated by a high-to-low transition on this line. To send data to the LCD, program should make sure that this line is low (0) and then set the other two control lines as required and put data on the data bus. When this is done, make EN high (1) and wait for the minimum amount of time as specified by the LCD datasheet, and end by bringing it to low (0) again.

When RS is low (0), data is treated as a command or special instruction by the LCD (such as clear screen, position cursor, etc.). When RS is high (1), data being sent is treated as text data which should be displayed on the screen.

written to the LCD.//



LCD Connections of Plug and Play board

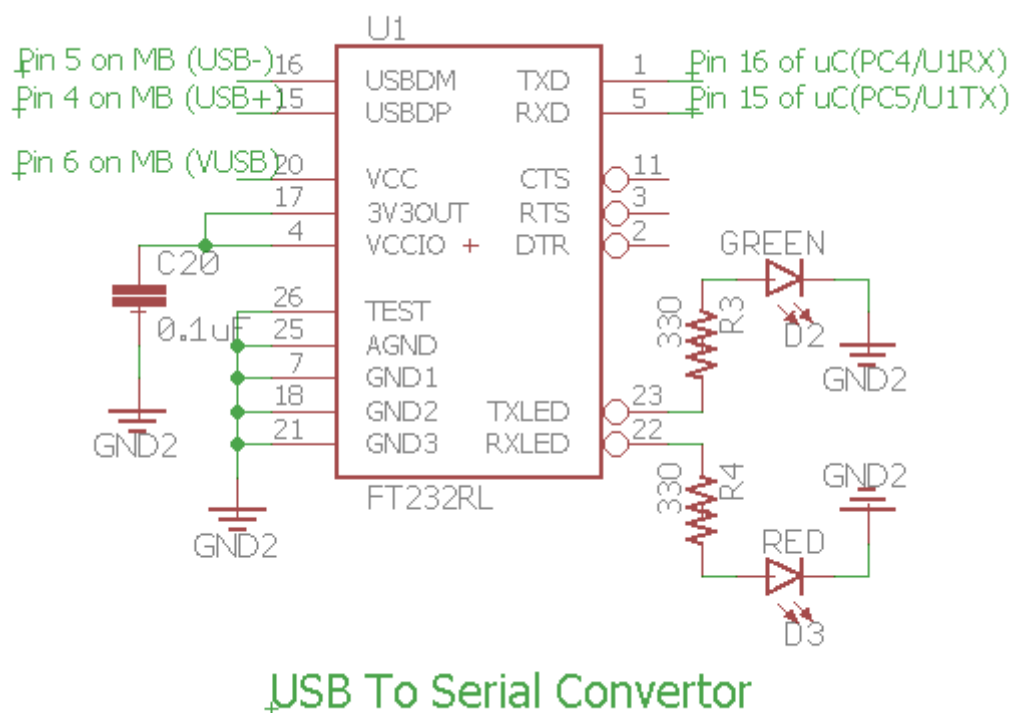


LCD Connections of uC based board

LCD	Pin Name(uC)	Pin Name(Plug and Play)
RS	PF0	PD6
EN	PF2	PD7
DB4	PD4	PA2
DB5	PD5	PA3
DB6	PD6	PA4
DB7	PD7	PA5

5.7 USB Communication

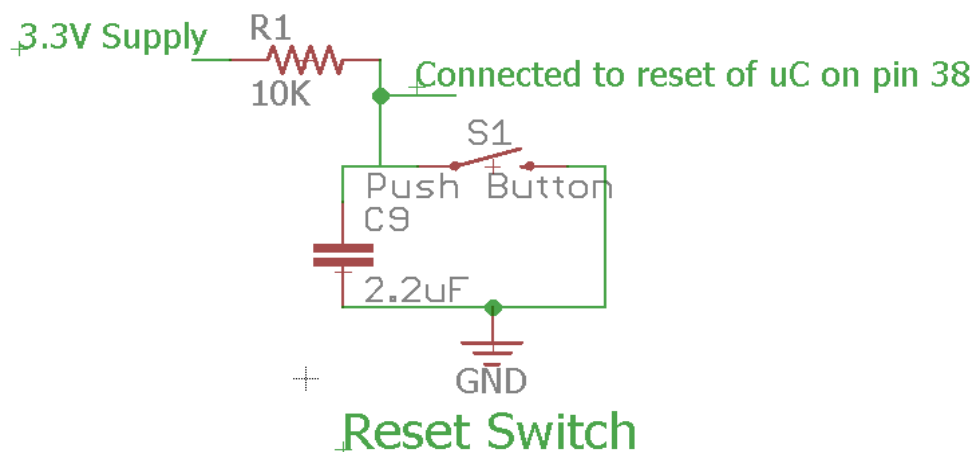
Fire Bird V's main board has USB port socket. Microcontroller accesses USB port via main board socket. All its pins are connected to the microcontroller adapter board via main board's socket connector. FT232 is a USB to TTL level serial converter. It is used for adding USB connectivity to the microcontroller adapter board. With onboard USB circuit Fire Bird V can communicate serially with the PC through USB port without the use of any external USB to Serial converter. Microcontroller socket uses USB port from the main board. Data transmission and reception is indicated using TX and RX LEDs which are located near the FT232 IC. This IC is only on the uC based board. Plug and play board has its own USB port on TIVA launchpad. The schematic of ft232 is shown below.



5.8 Programing the Controller

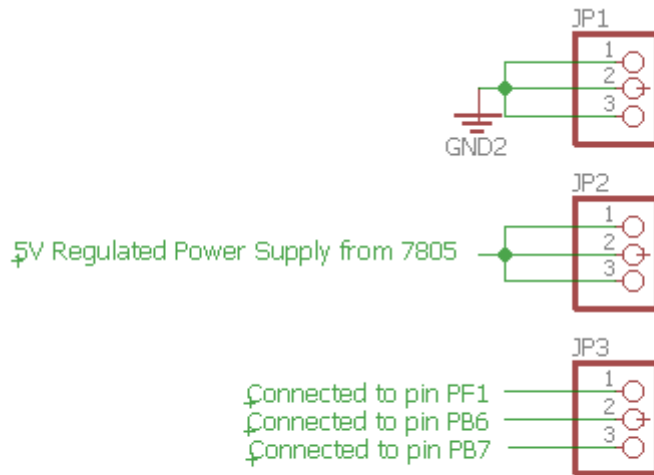
5.9 Reset Switch

The Plug and play board makes use of reset button present on the TIVA launchpad. The uC based has a switch connected to the reset the reset pin 38 of the microcontroller. The schematic is given below.

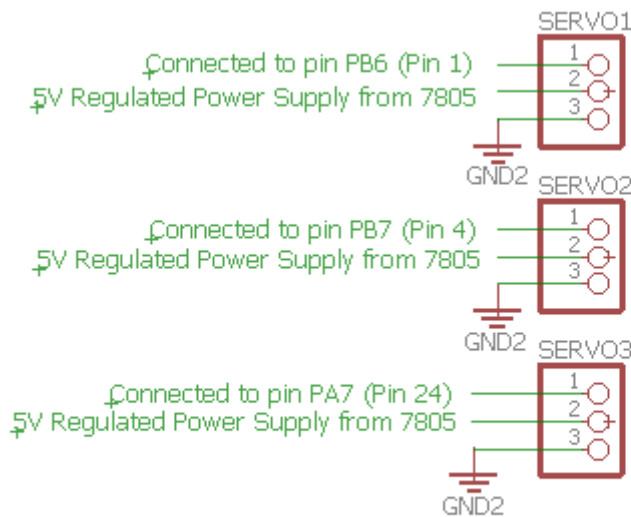


5.10 Servo Connectors

The microcontroller board has three Servo connectors. It can be used for driving servo motors of camera pod or any other attachment. Power for the servo connector is provided by the “5V servo supply” voltage regulator. Both the board have different pwm pins for servo which can be seen from the schematic.



Servo Connections for Plug and Play Board



Servo Connections for uC based Board

5.11 TM4C123GH6PM Micro-controller:

The Tiva™ C Series ARM Cortex-M4 microcontrollers provide top performance and advanced integration. The product family is positioned for cost-conscious applications requiring significant control processing and connectivity capabilities such as:

- Low power, hand-held smart devices
- Gaming equipment
- Home and commercial site monitoring and control
- Motion control
- Medical instrumentation
- Test and measurement equipment
- Factory automation
- Fire and security

- Smart Energy/Smart Grid solutions
- Intelligent lighting control
- Transportation

Schematic Of the connections is shown below.

5.12 Pin Functionality

6 Software Manual:

6.1 Code Composer Studio:

Code Composer Studio is an integrated development environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features. The intuitive IDE provides a single user interface taking you through each step of the application development flow. Familiar tools and interfaces allow users to get started faster than ever before. Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compelling feature-rich development environment for embedded developers. This description is directly taken from the website of Texas Instruments and click to know more [about CC Studio](#)

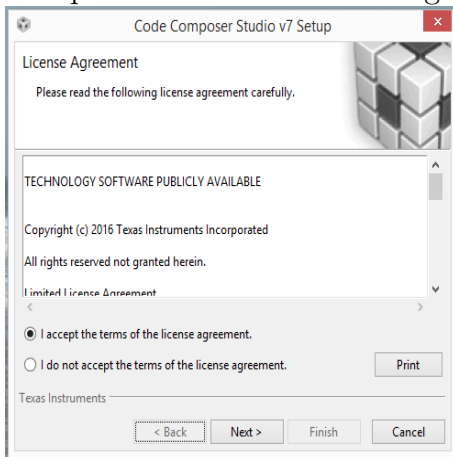
6.1.1 Download CC Studio:

At the time of writing this document Version 7 was the latest one. You can check for the latest at [Download CCS](#).(do not download any beta versions).There will be two installer files.The web installer will require Internet access until it completes. If the web installer version is unavailable or you can't get it to work, download, unzip and run the offline version. The offline download will be much larger than the installed size of CCS since it includes all the possible supported hardware.

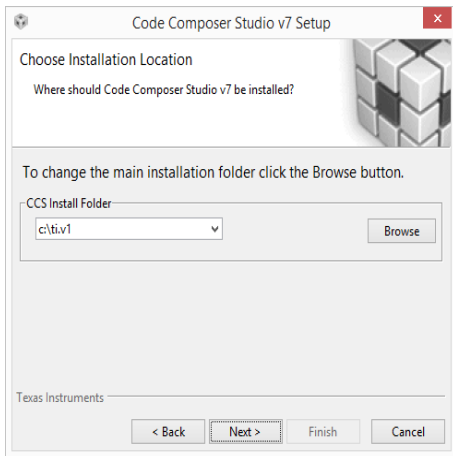
6.1.2 Installing C C Studio:

After the installer has started follow the steps mentioned below:

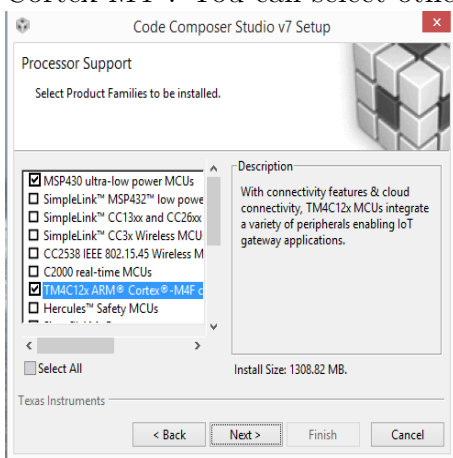
1. Accept the Software License Agreement and click Next.



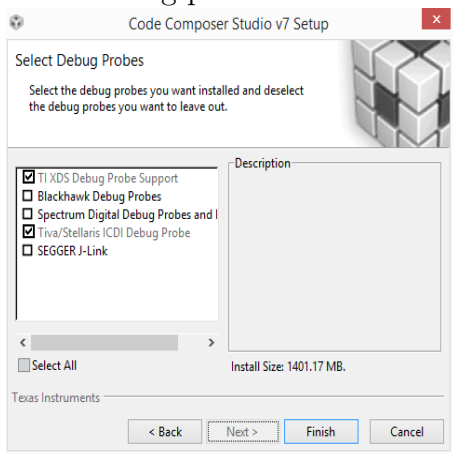
2. Select the destination folder and click next.



3. Select the processors that your CCS installation will support. You must select "TM4C12X Arm Cortex M4". You can select other architectures, but the installation time and size will increase.



4. Select debug probes and click finish



5. The installer process should take 15 - 30 minutes, depending on the speed of your connection. The offline installation should take 10 to 15 minutes. When the installation is complete, uncheck the "Launch Code Composer Studio v7" checkbox and then click Finish. There are several additional tools that require installation during the CCS install process. Click "Yes" or "OK" to proceed when these appear.
6. Install TivaWare for C Series (Complete). Download and install the latest full version of TivaWare from: [TivaWare](#). The filename is SW-TM4C-x.x.exe . This workshop was built using version 1.1. Your version may be a later one. If at all possible, please install TivaWare into the default location.

You can find additional information at these websites:

Main page: www.ti.com/launchpad

Tiva C Series TM4C123G LaunchPad:

<http://www.ti.com/tool/ek-tm4c123gxl>

TM4C123GH6PM folder:

<http://www.ti.com/product/tm4c123gh6pm>

BoosterPack webpage: www.ti.com/boosterpack

LaunchPad Wiki:

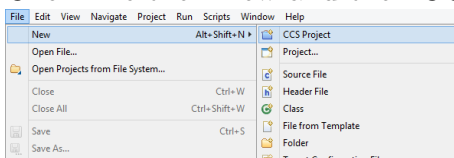
www.ti.com/launchpadwiki

For understanding the launchpad properly and to learn more about Tiva it is strongly recommended to go through the webpage **Tiva Workshops** and download and read the workbook

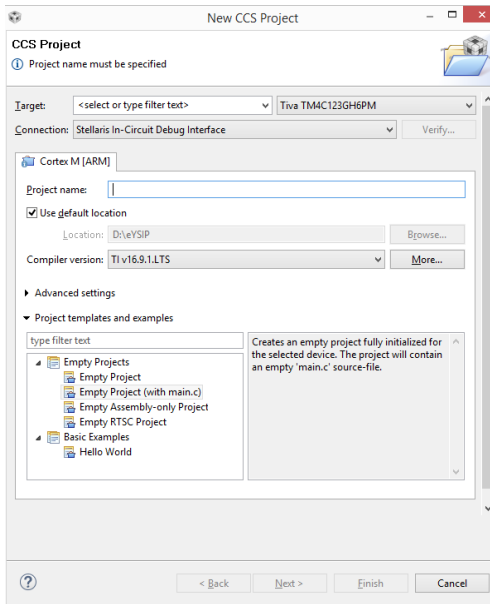
6.1.3 Create a New Project

To create new project follow the steps mentioned:

1. Click File then New and then CCS Projects



2. Select Target and connection as shown in the photo. Give a name to your project and save in a location. Click Finish. A main.c file will be open



6.1.4 Add Path and Build Variables

The path and build variables are used for:

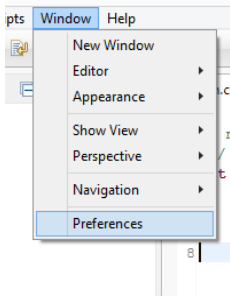
- Path variable – when you ADD (link) a file to your project, you can specify a "relative to" path. The default is PROJECT_LOC which means that your linked resource (like a .lib file) will be linked relative to your project directory.
- Build variable – used for items such as the search path for include files associated with a library – i.e. it is used when you build your project.

Variables can either have a PROJECT scope (that they only work for this project) or a WORKSPACE scope (that they work across all projects in the workspace). In the next step, we need to add (link) a library file and then add a search path for include files. First, we'll add these variables MANUALLY as WORKSPACE variables so that any project in your workspace can use the variables. Refer to the workbook by TI for adding as PROJECT

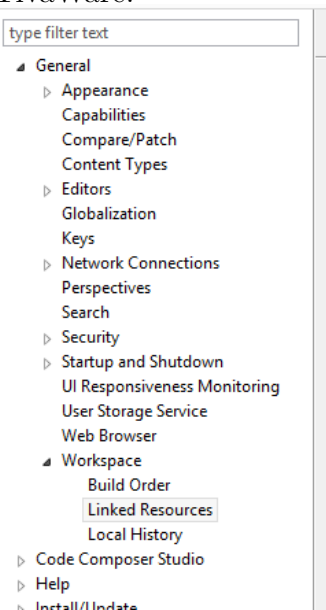
6.1.4.1 Adding a Path Variable

To add a path variable,:

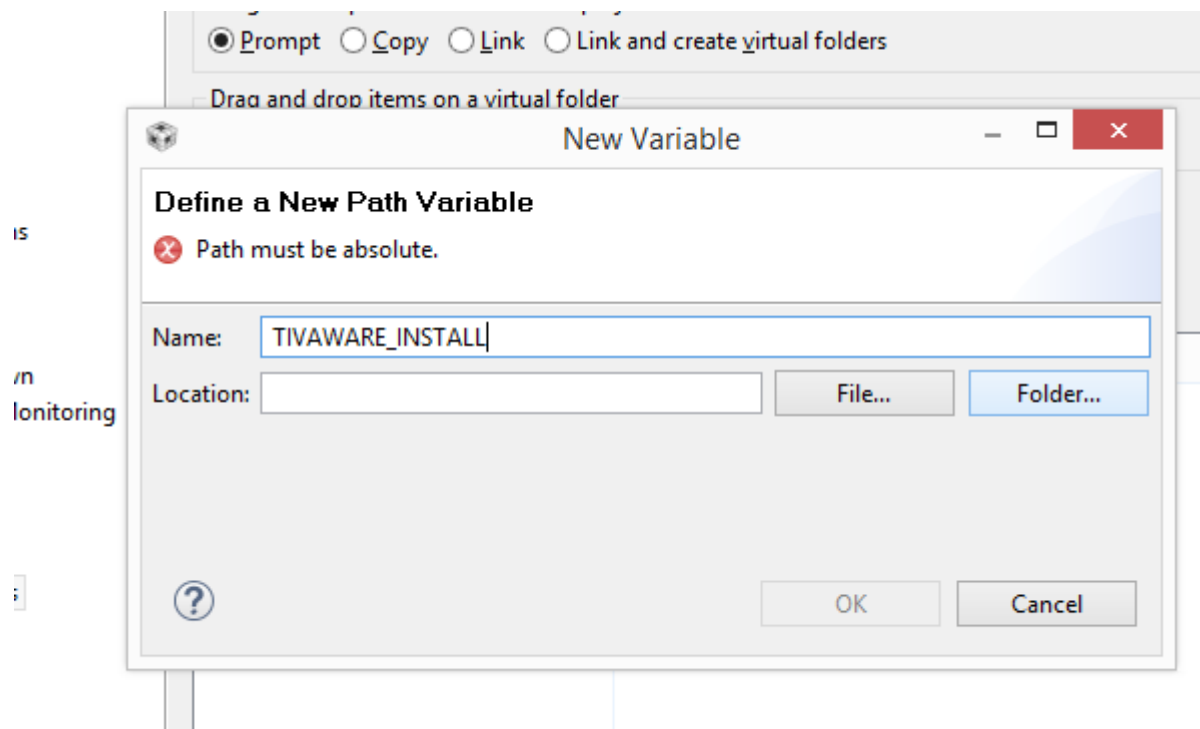
- Right-click on your Window Tab and select Preference.



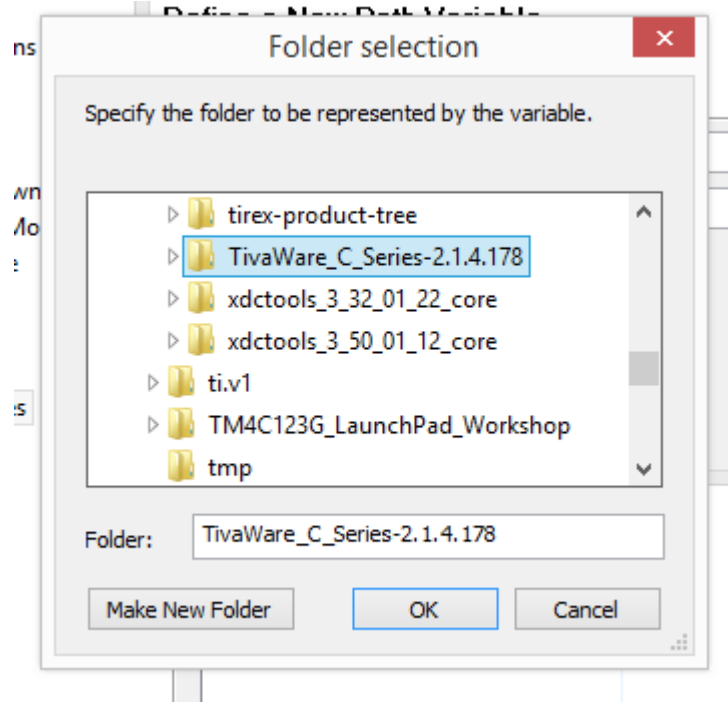
- Expand General list in the upper left-hand corner as shown and then expand the Resource list and click on Linked Resources: We want to add a New variable to specify exactly where you installed TivaWare.

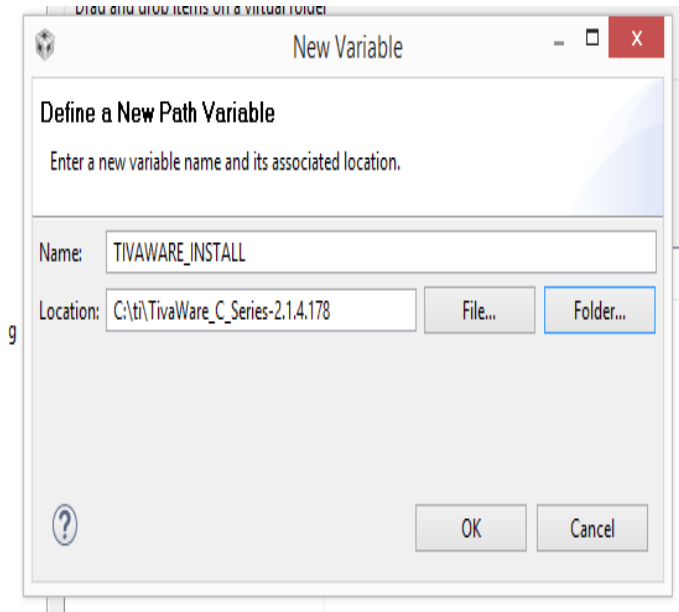


- Click New
- When the New Variable dialog appears, type TIVAWARE_INSTALL for the name.

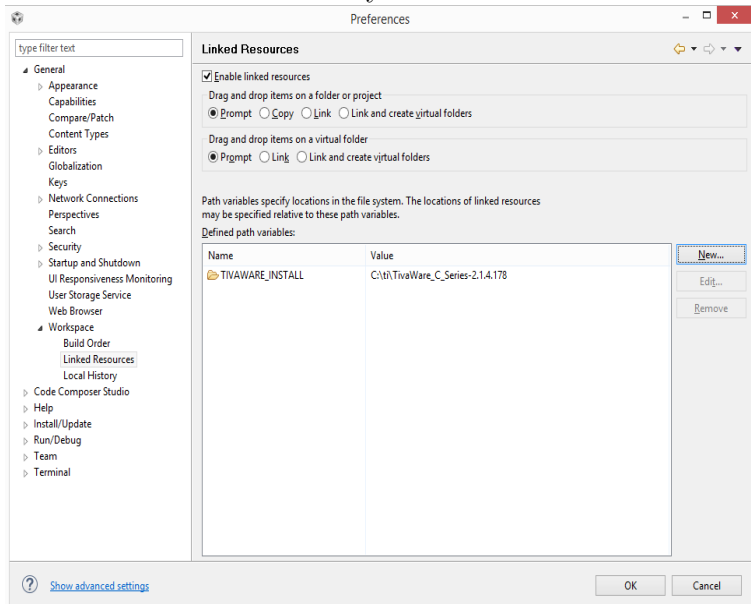


- For the Location, click the Folder... button and navigate to your TivaWare installation. Click on the folder name and then click OK.





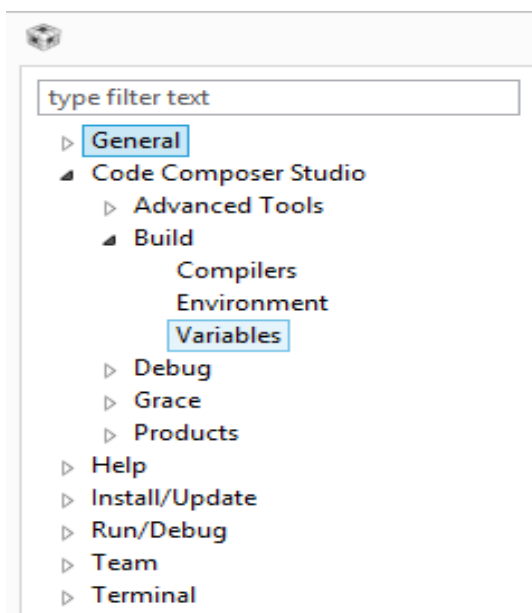
- Click OK. You should see your new variable listed in the Variables list.



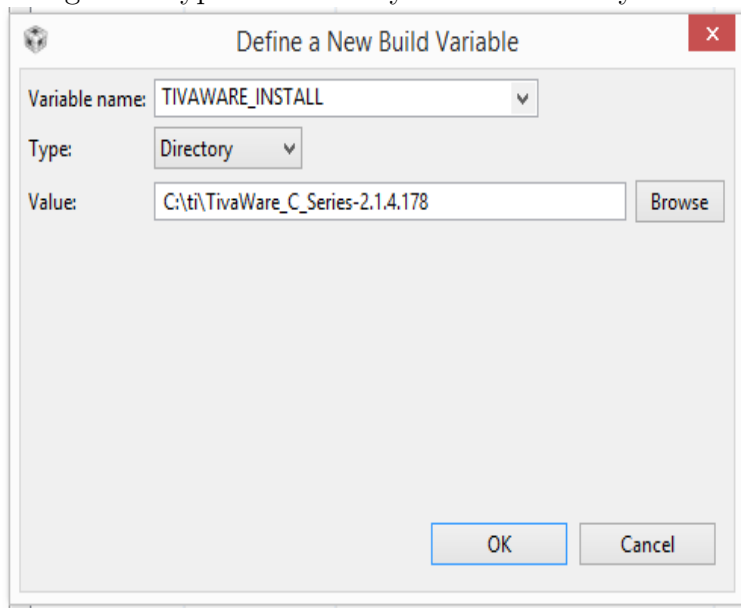
6.1.4.2 Adding a Build Variable

Now let's add a build variable that we will use in the include search path for the INCLUDE files associated with the TivaWare driver libraries.

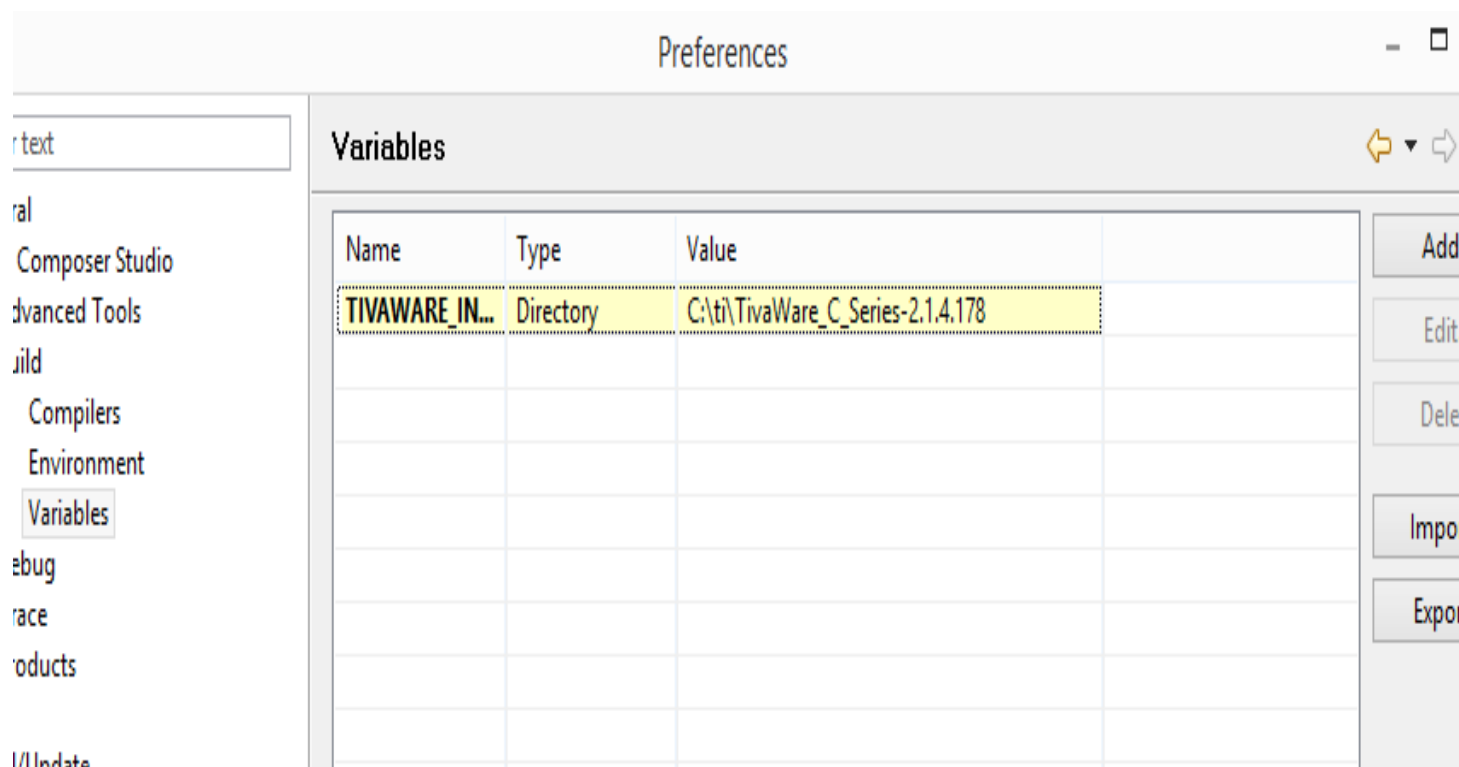
- Click on Code Composer Studio Build and then the Variables tab:



- Click the Add button. When the Define a New Build Variable dialog appears, insert TIVWARE_INSTALL into the Variables name box.
- Change the Type to Directory and browse to your Tivaware installation folder.



- Click OK.
- Click OK again to save and close the Build Properties window.



6.2 driver.lib

6.3 Buzzer

Located in the folder “Buzzer.Beep” folder in the documentation. In this example, we will load buzzer beep code in Tiva based Fire Bird V. Now we will see in detail the structure of this code. This experiment demonstrates the simple operation of Buzzer ON/OFF with one some delay. Buzzer is connected to PORTF 4 pin of the Tiva Launchpad. If you have uC based board then it is connected to PORTA 2. Concepts covered: Output operation, generating delay Note: Make sure that you have included driver.lib // Buzzer is connected at PF4/PA2 on Tiva launchPad/uC // To turn it on make PF4/PA2 pin logic 1

6.4 **Programming the Robot**

6.5 **Using Debugger of The Programmer**

6.6 **Simple I/O Operation**

6.7 **Robot Direction Control**

6.8 **Robot Position Control Using Interrupts**

6.9 **Timers and its Interrupts**

6.10 **Robot Speed Control**

6.11 **LCD Interfacing**

6.12 **Analog To Digital Converter**

6.13 **Serial Communication**

6.14 **I2C Communication**